1981-12

An investigation of a steady state, allocation model for military manpower planning

Lott, Joseph Nathaniel

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

AN INVESTIGATION OF A STEADY STATE,
ALLOCATION MODEL FOR MILITARY MANPOWER PLANNING

by

Joseph Nathaniel Lott

December 1981

Advisors: P. R. Milch and
K. T. Marshall

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Investigation of a Steady State, Allocation Model for Military Manpower Planning	5. TYPE OF REPORT & PERIOD COVERED Master's Thesis December 1981
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Joseph Nathaniel Lott	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	12. REPORT DATE December 1981
	13. NUMBER OF PAGES 85
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Allocation models, manpower models, manpower planning, goal programming, elastic programming, quadratic programming.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The problem of allocating different types of jobs among several classes of military manpower is becoming increasingly more important as military jobs grow more specialized. A model was proposed by Richard C. Grinold which constructs a personnel inventory by rank for each of several classes of manpower and then allocates that inventory to meet billet requirements. The model is designed for long-range planning

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

purposes and produces possible inventory based on an optimization scheme that sets permitted errors in the allocation. The thesis presents a review and demonstration of the model based on the U.S. Navy officer corps, a discussion of implementation considerations, and further work on optimization schemes.

An Investigation of a Steady State, Allocation Model for Military Manpower Planning

by

Joseph Nathaniel Lott
Captain, United States Marine Corps
B.I.E., Auburn University, 1976

Submitted in partial fulfillment of the
Requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
December 1981

ABSTRACT

The problem of allocating different types of jobs among several classes of military manpower is becoming increasingly more important as military jobs grow more specialized. A model was proposed by Richard C. Grinold which constructs a personnel inventory by rank for each of several classes of manpower and then allocates that inventory to meet billet requirements. The model is designed for long-range planning purposes and produces possible inventory based on an optimization scheme that sets permitted errors in the allocation. The thesis presents a review and demonstration of the model based on the U.S. Navy officer corps, a discussion of implementation considerations, and further work on optimization schemes.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

The problem of allocating different types of jobs among several classes of military manpower is becoming increasingly more important as military jobs become more specialized. A model was proposed by Grinold [Ref. 1] which constructs a personnel inventory by rank for each of several classes of manpower and then allocates that inventory to meet billet requirements. The model is designed for long-range planning purposes and produces possible inventories based on an optimization scheme that sets permitted errors in the allocations.

The next section reviews the model and offers several points to consider concerning its implementation based on practical problems during the preparation of this thesis. The review highlights the mathematical foundations of the model and defines the essential variables and parameters. A complete discussion of the mathematical concepts is found in Grinold [Ref. 1].

Section III presents a means for measuring the error between requirements and inventories and then expresses this measure as the objective of 3 types of mathematical programs: linear (goal) programming, elastic programming, and quadratic programming. Sections IV, V, and VI present a variety of possible formulations for the three programming models from which analysis may be accomplished. The formulations presented include those proposed by Grinold [Ref. 1] plus several new ones. For the quadratic programming models, analytical solutions are derived from which computer programs in the APL language were written.

Section VII is a demonstration of the model using both APL programs and a linear programming optimizer available at the Naval Postgraduate School. The demonstration is based on typical data from the U.S. Navy officer corps. The last section summarizes the results of the investigation and proposes areas for further studies.

II. REVIEW OF THE MODEL

A. PURPOSE

The purpose of the model is to allocate different types of jobs among several classes of manpower in a steady state system. The term "allocation" refers to a method for distributing jobs among several classes of people or people among different types of jobs. The allocations are determined by arrays of sharing fractions which are defined below. These sharing fractions represent an actual or possible management policy.

The term "steady state" refers to the equilibrium condition prevailing in the model. This condition arises from the assumption of a fixed longitudinal manpower flow in each manpower class used for the analysis. This concept is defined for the model in Grinold [Ref. 1], and is discussed in detail in Grinold and Marshall [Ref. 2], and in Bartholomew and Forbes [Ref. 3]. Further discussion of the steady state conditions is found in the section which defines the steady state variables.

The final item of interest in stating the purpose deals with the planning aspects of the model. Since this analysis is done for steady state conditions, it can be used most effectively to test current or alternate management policies for their long-term effect and provide answers to the "what if" policy questions. As an aggregate planning tool, the model is intended to test long-term policies that may be used later for daily operation of a manpower system. Since conditions of steady state may be significantly different from current conditions,

the analyst should be aware of the limitations of the model. Often, the relative differences between competing policies or the magnitude of the effect of a policy may be the point of interest for which an analysis is being performed. The model should be used with these considerations in mind.

The model is motivated by a study of the U.S. Navy officer corps and examples from this manpower system will be used when explaining concepts as an aid to better understanding of the model. A description of the structure of the U.S. Navy officer corps for the unrestricted line officers is found in NAVPERS 15197A [Ref. 4].

B. DEFINITIONS AND TERMINOLOGY

1. Attributes

The definition and understanding of variables in the model is complicated by the large number of variables present. The most important variables relating to concepts are presented in this section. Other variables relating to specific optimization techniques or used for mathematical clarity will be introduced as the need for them arises. A list of variables used in the model is found in Appendix A.

The requirements and the inventories possess three common attributes. They are:

- MANPOWER CLASSES--the types of manpower; indexed by $k=1,2,\ldots,K$.
- CAREER STAGES--the experience levels required; indexed by $i=1,2,\ldots,I$.
- JOB CLASSES--the types of jobs to be filled; indexed by $j=1,2,\ldots,J$.

The clear definition of attributes is closely related to the purpose of analysis. In the demonstration which follows, the manpower classes

represent types of officers (e.g., Pilot, Naval Flight Officer); the career stages represent ranks (e.g., Lieutenant, Commander); and the job classes are naval officer billet categories (e.g., 1310, 1320).

The level of detail is also closely related to the purpose of analysis. For example, the manpower class of pilot could be further broken down into pilots by type of aircraft, but this imposes increased data requirements which may not be needed if the purpose of analysis is to study Navy-wide trends rather than issues specific to the Navy's aviation community.

The attributes may apply to either requirements or inventories. In Navy terminology, requirements are usually stated in terms of "billets." Some confusion may arise since this is synonymous with the term "job." The reader should bear in mind that requirements may be referred to as billets or jobs which are distinct from the attribute of "job type." Likewise, the term "inventories" is used interchangeably with stocks, personnel, and people, but is distinctly different from the term "manpower types." The use of these terms should be clear from the context in which they occur.

The demonstration highlights another problem found in practical application. The types of people and types of jobs are often expressed by the same terminology. For example, 1310 is both an officer designator and a billet designator for the manpower class "pilot" and for the job class "pilot." To avoid confusion, the word equivalent "pilot" will be used when speaking of 1310 officers and the billet designator "1310" will be used when speaking of a corresponding type of job.

Other interpretations of the model based on current military personnel problems are possible. For example, manpower classes could be defined as enlisted personnel by mental group category and the purpose of analysis could be to study the effects on billets of increasing requirements for the higher mental groups with a shrinking manpower pool from which to provide accessions. Another possibility would be to define manpower classes by sex and job classes by types of sea or shore billets and then study the implications of increased accession of women on sea-shore rotation cycles.

2. Steady State Variables

Let n_i be the number of time periods an individual must remain in the organization before entering stage (i+1). For example, naval officers in the rank of lieutenant (jg) will remain in the Navy four years before being promoted to the rank of lieutenant, thus n_i for lieutenant (jg) is four. The total amount of time a person may spend in stage i is:

$$n_i - n_{i-1} \qquad (n_0 = 0).$$

An example of the amount of time spent in each career stage of the Navy officer corps will be given in Section VI (Fig. 4) for the demonstration. The steady state variables are defined as follows:

- Survivor fractions by rank and manpower class--this is the matrix S indexed over (I,K). The $(i,k)^{th}$ element, s_{ik}, represents the fraction of type k people who remain in the organization more than i stages. The element s_{ik} may also be interpreted as the probability that a type k person remains in the organization longer than n_i time periods.

- Expected stage length is the matrix W whose $(i,k)^{th}$ element represents the amount of time a person in manpower class k is expected to spend in stage i. W may also be defined as the product of the probability of a type k person reaching stage i and the length of stage i:

$$w_{ik} = S_{ik}(n_i - n_{i-1}).$$

A more complete discussion of survivor fractions and expected stage lengths may be found in Grinold [Ref. 1] and Grinold and Marshall [Ref. 2].

3. Requirements

Requirements are expressed in three ways:

- Billets by rank and job type--the number of jobs at each stage to be filled. This is a matrix B whose elements are indexed by (i,j).
- Billets by rank and people type--the number of people required at each stage. This is not the normal way in which requirements are stated and should not be confused with personnel inventories. This is a matrix P whose elements are indexed by (i,k).
- Target allocations--the number of (i,k) people desired in (i,j) jobs. This is a three-dimensional array T with elements indexed by (k,i,j).

4. Inventories and Accessions

Finding optimal inventories and accessions will normally be the goal of this model. They are defined below.

- Accessions are denoted by a vector Y whose k^{th} element represents the number of people entering manpower class k each year.

Inventories are expressed in three ways:

- Inventories by rank and job type--the matrix X whose elements are indexed by (i,j) defines the number of jobs at each stage actually filled. This is not the normal way in which inventories are stated and should not be confused with billet requirements.
- Inventories by rank and people type--the actual number of available people by manpower class and rank. This is the matrix Z whose elements are indexed by (i,k).
- Actual allocation--the actual number of (i,k) people who will fill (i,j) jobs. This is the three-dimensional array A with elements indexed by (k,i,j). The notation for requirements and inventories is summarized in Table 1.

TABLE 1

REQUIREMENTS AND INVENTORIES

	Job Type and Rank	People Type and Rank	People Type, Rank and Job Type
Requirements	b_{ij}	p_{ik}	t_{kij}
Inventories	x_{ij}	z_{ik}	a_{kij}

5. Sharing Rules

There are two rules used to distribute the (i,j) jobs among the (i,k) manpower types or vice versa. These rules are known as job sharing and people sharing.

- Job sharing is defined by a three-dimensional array F whose $(k,i,j)^{th}$ element gives the fraction of (i,j) jobs that should be performed by manpower type k.
- People sharing is defined by a three-dimensional array G whose $(k,i,j)^{th}$ element gives the fraction of (i,k) manpower types who should perform job type j.

C. DETERMINATION OF REQUIREMENTS

The target allocation (T), which is the number of type j jobs to be filled by class k people of rank i, is determined in one of two ways:

1. By specifying the desired billets by people type and rank (P) and the rule for sharing these people among the different billets (G). Note that G times P distributes the requirement for the k^{th} class of people in rank i over the job class j and produces the target allocation T:

$$g_{kij}\ p_{ik} = t_{kij}.$$

2. By specifying the billets by rank and job type (B) and the rule for sharing these billets among the different types of people (F). Note that F times B distributes the j^{th} type of job in rank i over the people class k and also produces the target allocation T:

$$f_{kij}\ b_{ij} = t_{kij}.$$

In order to maintain consistency, the following equation must hold:

$$g_{kij}\ p_{ik} = t_{kij} = f_{kij}\ b_{ij} \qquad (1)$$

The argument from which this type of equation is derived will be referred to as the "consistency rule." Furthermore, all people classes must be filling some job and all job classes must be filled by some people which means that the equation:

$$\sum_j g_{kij} = 1$$

implies that

$$p_{ik} = \sum_j f_{kij} \, b_{ij} \tag{2}$$

and the equation:

$$\sum_k f_{kij} = 1$$

implies

$$b_{ij} = \sum_k g_{kij} \, p_{ik} \tag{3}$$

Thus, one may either start with F and B and calculate P from equation (2) and then G using equation (1), or start with G and P and calculate B from equation (3) and then F from equation (1).

For the Navy example which follows, it should be noted that the usual manner of determining target allocations will be to start with a given requirement of job types by each rank (B) derived from the force structure. Since it is more common to think of people being assigned to jobs rather than jobs to people, a typical sharing policy is more easily understood in terms of distributing people over the required job types (G). Thus, it would be helpful to be able to calculate F and P based on B and G. Unfortunately, this cannot be done directly. One solution to this problem is a least squares approach using equation (3). First determine the least squares fit values for P, and then solve for F using equation (1). The least squares objective is:

$$\min_P \sum_i \sum_j \left(\sum_k g_{kij} \, p_{ik} - b_{ij} \right)^2 .$$

Notice that this minimization is accomplished by minimizing each of the following I expressions:

$$\min_{P} \sum_{j} \left(\sum_{k} g_{kij} \, p_{ik} - b_{j}\right)^{2} \quad \text{for } i\text{-}1,2,\ldots,I.$$

For a particular i, the above can be written in matrix notation:

$$\min \, (G'P - B)' \, (G'P - B).$$

Thus, the i^{th} column of P is given by the optimal solution:

$$P = [G \; G']^{-1} \, G \; B \; .$$

Now calculate f using equation (1) to obtain:

$$f_{kij} = \begin{cases} g_{kij} \, p_{ik} \, (b_{ij})^{-1} & \text{if } b_{ij} \neq 0, \\ 0 & \text{if } b_{ij} = 0. \end{cases}$$

D. DETERMINATION OF INVENTORIES

The actual allocation (A), which is the number of people of type k in rank i who will be available to fill job type j, is determined using the same consistency rule invoked for equation (1):

$$g_{kij} \, z_{ik} = a_{kij} = f_{kij} \, x_{ij} \quad , \tag{4}$$

and the implications similar to equations (2) and (3) above must also hold. The equation:

$$\sum_{j} g_{kij} = 1$$

implies that

$$z_{ik} = \sum_{j} f_{kij} \, x_{ij} \; , \tag{5}$$

and the equation:

$$\sum_{k} f_{kij} = 1$$

implies

$$x_{ij} = \sum_{k} g_{kij} \, z_{ik} \; . \tag{6}$$

In addition, the model is constrained by the steady state requirement that the expected inventory of type k people in rank i is equal to the number of accessions of type k people times the expected duration of type k people in rank i. In equation form, this constraint is:

$$z_{ik} = w_{ik}\, y_k \,. \tag{7}$$

Since inventories may be computed from accessions (Y) and expected stage lengths (W), the optimization techniques to determine inventories found in the next sections will usually have Y as the decision variable. If the inventories (X or Z) are the decision variables, then a method is needed to determine the corresponding accessions. From equation (7), note that for each y_k there are I equations. Once again, a least squares approach may be used to find the best fit values of y_k for this system of equations. The objective is:

$$\min_{y} \sum_{i} (w_{ik}\, y_k - z_{ik})^2 \,.$$

This yields the result:

$$y_k = \left(\sum_i w_{ik}\, z_{ik}\right) \left(\sum_i w_{ik}^2\right)^{-1} \quad \text{for } k=1,2,\ldots,K.$$

III. OPTIMIZATION TECHNIQUES

A. PURPOSE

This section discusses the use of mathematical programming to determine desired values for the variables defined in Section II. The degree to which these techniques may be used depends on availability of data, level of detail in attribute definition, and desired accuracy of results. The scope of the problem to be solved determines the type of the optimization technique and the way in which the problem is formulated.

The purpose of this section is to present a method by which the quality of an allocation may be measured. Such a measure will be used as the basis for the objective functions for three types of mathematical programs: linear (goal), elastic, and quadratic programming. Formulations for each of these will be presented in subsequent sections.

B. FORMULATION OF GOALS

In formulating the optimization, special care must be taken in determining the objective to be achieved and the decision variable to be determined. The overall objective is to match actual allocations to target allocations with respect to a decision variable. The methods described here seek to do this by minimizing the weighted percent difference between inventories and requirements. The decision variable to be determined may be any one of the previously defined variables. All other variables are then treated as parameters.

Listed below are the possible decision variables and a brief discussion of how they may affect the optimization.

- Requirements--Errors may be minimized with either billets by job type and rank (B) or billets by people type and rank (P) as the decision variable. Normally B is treated as a parameter derived from the force structure. For example, if ten aviation squadrons of a certain type are required by the force structure, then various numbers of officer pilots of different ranks will be needed to fill the aviation type billets found in those squadrons. On occasion, it may be of interest to allow the requirements to be decision variables and check the resulting optimal values given fixed inventories and sharing rules.
- Inventories and Accessions--The decision variable may be either the inventories or accessions. Due to the steady state assumption, it is sufficient to know the accessions and the expected length of time in each career stage in order to calculate expected inventories. For this reason, the accessions (Y) will normally be the decision variables. This approach requires that billets (B or P), sharing rules (F or G), and expected duration in each stage (W), be treated as parameters. The objective functions in Grinold [Ref. 1] are of this type.
- Expected Duration in Each Career Stage--This variable may be the decision variable if billets, accessions and sharing rules are known. If certain assumptions are made about the conditions from which an optimal W is derived, then it is possible to relate the optimal W to a survivor function.
- Sharing Rules--Minimizing errors with F or G as the decision variable will result in the optimal "mixing" policy. If the

objective is to find the optimal policy based only on billets, inventories and expected stage lengths, then this approach is reasonable.

C. MEASUREMENT OF ERROR

As stated previously, optimization is performed to minimize the weighted error between requirements and inventories. The error for the linear models and quadratic models may be expressed as the excess or deficit of the three quantities shown in Table 2.

TABLE 2

TYPES OF ERROR

$\lvert t_{kij} - a_{kij}\rvert$	:	the error between targets and allocations by job type, rank, and people type.
$d_{ik} = \lvert z_{ik} - p_{ik}\rvert$	:	the error between the inventory of (i,k) people and the corresponding billets.
$e_{ij} = \lvert b_{ij} - x_{ij}\rvert$	:	the error between the (i,j) billets and the corresponding inventory.

The first error shown involves all three attributes and may be obtained from either the second or the third one since these errors may be distributed using the job sharing or people sharing rule:

$$g_{kij}\, d_{ik} = \lvert t_{kij} - a_{kij}\rvert = f_{kij}\, e_{ij}$$

where d_{ik} and e_{ij} are defined in Table 2.

For this reason, the last two types of errors, those by rank and job type and those by rank and people type, will be used in the following discussion. In the linear programming model, it is important to know if

the error is a shortfall or overfill. In keeping with the notation in Lee [Ref. 5], shortfalls will be denoted by the superscript minus and overfills by the superscript plus.

Weighting factors are used to allow trade-offs between billets of differing attributes. The weighting factors are based on total number of billets to be filled and the relative importance of billets. A shortfall of twenty billets of a particular type will have a different impact depending on whether there were 100 or 1,000 billets to be filled. The error values will have a common metric if they are converted to percent errors. Thus, a better measure of the error would be $(z-p)/p$ or $(x-b)/b$. However, the actual measure may still differ depending on the nature of the job type, rank, or personnel type. For example, a five percent shortfall of pilots who are of the rank captain may not be as serious as a five percent shortfall in pilots who are lieutenants. Thus the percent shortfall in meeting a critical target should be weighted more heavily than that of a less critical target. In order to handle this problem, additional parameters indicating the percent trade-off between ranks, job types, and people types must be introduced to the model. One way in which this may be done is to specify parameters which will serve as benchmarks to measure the permitted range of unit error in overfilling or underfilling a requirement.

For example, let a five percent unit error be the permitted shortfall of 1310 (pilot) lieutenant jobs. Now for another job, say 1310 captains, the relative trade-off for an underfill may be found by answering the question, "What percent under target in the assignment of 1310 captain billets is as serious as a five percent shortfall in filling 1310

lieutenant billets?" In this way, essential judgments about the trade-off between differing ranks and job types may be made. The same process could also be done for ranks and people types or the consistency equation could be used to produce the trade-off parameters. The notation for these parameters is shown in Table 3. These trade-offs may now be added to the penalty function by multiplying the percent error by the inverse of the permitted unit error. Thus the actual weighting factors for each error is the inverse of the permitted unit percent error times the requirements. Let u and v be the weighting factors for job types and people types, respectively. As before, let the superscript plus denote weights associated with overfills and the superscript minus denote weights associated with shortfalls. The notation for these parameters is in Table 4. The total penalty value is now defined as the weighting factor times the error. This penalty value is a measure of error that takes into account both the varying sizes of the different requirements and their relative importance. The notation for the penalty measure derived thus far is in Table 5.

Obtaining valid weights is of particular importance to the model since the final result of the optimization could be very sensitive to these parameters. There are several ways in which they can be estimated. Expert judges could arbitrarily select either the parameters to be used or the range of minimal values for excess and deficit. For example, the requirement for a particular job type and rank may be 100. If a permitted range of error is 95 to 110, then ϕ = 5 percent and θ = 10 percent. The range of error is a useful concept and the range bounds $\bar{b}$ $\underline{b}$, $\bar{p}$ and $\underline{p}$ are defined in Table 6. If the lower unit errors are greater than 100 percent, then it would be possible to have a negative value for the lower

TABLE 3

PERMITTED UNIT PERCENT ERROR

	by job type and rank	by people type and rank
shortfall	ψ_{ij}	δ_{ik}
overfill	θ_{ij}	ϕ_{ik}

TABLE 4

WEIGHTING FACTORS

	by job type and rank	by people type and rank
shortfall	$u_{ij}^{-} = (\psi_{ij}\ b_{ij})^{-1}$	$v_{ik}^{-} = (\delta_{ik}\ p_{ik})^{-1}$
overfill	$u_{ij}^{+} = (\theta_{ij}\ b_{ij})^{-1}$	$v_{ik}^{+} = (\phi_{ik}\ p_{ik})^{-1}$

TABLE 5

WEIGHTED PENALTY MEASURE

	by job type and rank	by people type and rank
shortfall	$u_{ij}^{-}\ e_{ij}^{-} = \dfrac{b_{ij} - x_{ij}}{\psi_{ij}\ b_{ij}}$	$v_{ik}^{-}\ d_{ik}^{-} = \dfrac{p_{ik} - z_{ik}}{\delta_{ik}\ p_{ik}}$
overfill	$u_{ij}^{+}\ e_{ij}^{+} = \dfrac{x_{ij} - b_{ij}}{\theta_{ij}\ b_{ij}}$	$v_{ik}^{+}\ d_{ik}^{+} = \dfrac{z_{ik} - p_{ik}}{\phi_{ik}\ p_{ik}}$

TABLE 6

ERROR RANGE OF REQUIREMENTS

	by job type and rank	by people type and rank
lower error bound	$(1 - \psi_{ij})\, b_{ij} = \underline{b}_{ij}$	$(1 - \delta_{ik})\, p_{ik} = \underline{p}_{ik}$
upper error bound	$(1 + \theta_{ij})\, b_{ij} = \bar{b}_{ij}$	$(1 + \phi_{ik})\, p_{ik} = \bar{p}_{ik}$

bound. To avoid this possibility, it will be required by definition that the lower bounds be greater than or equal to zero. It is also possible to estimate the permitted unit error by job type if given people type and vice versa using the consistency rule resulting in the equations:

$$\underline{p}_{ik}\, g_{kij} = \underline{b}_{ij}\, f_{kij} \text{ for all } k,i,j;$$

$$\bar{p}_{ik}\, g_{kij} = \bar{b}_{ij}\, f_{kij} \text{ for all } k,i,j.$$

D. OBJECTIVE FUNCTIONS

1. General

Let E_{ij} be the penalty value function of errors in job class j and rank i, and let D_{ik} be the penalty value function of errors in people class k and rank i. Then three possible mathematical programming objective functions would be:

(1) $\min \sum_i \sum_j E_{ij}$, i.e., minimizing error by job types;

(2) $\min \sum_i \sum_k D_{ik}$, i.e., minimizing error by people types;

(3) $\min \sum_i \sum_j E_{ij} + \sum_i \sum_k D_{ik}$, i.e., minimizing error by job types and people types.

Minimizing the third objective function should produce a close match between targets and actual allocations and was used in the mathematical programs in

Grinold [Ref. 1]. The use of objective (3) requires more data input for parameters than either objectives (1) or (2). Objective functions based on each of the three representations will be used in the following sections.

In order to shorten mathematical notation in the following sections on specific objective functions, it will be assumed that the decision variable of interest is the inventory X and the objective function is type (1) from above. Also, the subscripts will be dropped for clarity.

2. Linear (Goal) Programming

One objective function that measures the weighted percent errors is a piecewise linear convex function that has a value of zero if the actual allocation meets the target, and increases at a rate inversely proportional to the permitted error in meeting that target. This means the function is unity when the decision variable assumes either the lower or the upper error bound. Therefore, the function will satisfy the following three conditions:

(i) $E(x) = 0$ if $x = b$,

(ii) $E(x) = 1$ if $x = \underline{b}$,

(iii) $E(x) = 1$ if $x = \bar{b}$.

Such a function, as shown in Figure 1, is:

$$E(x) = \max \quad \frac{(x - b)}{\theta b}, \frac{(b - x)}{\psi b}$$

$$= \max\,(u^{+} e^{+}, u^{-} e^{-}) .$$

Notice that the penalty value will be relatively small (less than 1) for values between $\underline{b}$ and $\bar{b}$ and that it will increase at a rate $1/(b\ \psi)$ if $x < b$ or $1/(b\ \theta)$ if $b < x$.

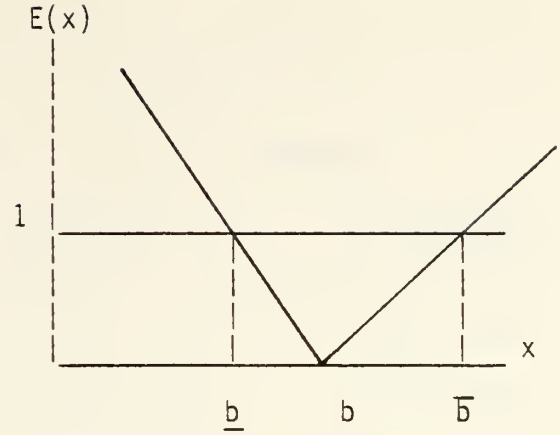

Fig. 1. Piecewise Function for Linear Programming.

3. Elastic Programming

For this type of programming model, the definition of the error will be modified slightly as shown in Table 7. The weighting factors will be changed since the actual error is now slightly different due to the use of the error bounds rather than the actual requirements. In order to notationally indicate this difference, the use of underscore and overscore for weights and errors will be used in place of superscript plus and minus. This type of programming can use a piecewise linear objective function of the following type:

$$E(x) = \begin{cases} \dfrac{(\underline{b} - x)}{\psi \, \underline{b}} & \text{if } x < \underline{b} \,, \\ 0 & \text{if } \underline{b} \leq x \leq \overline{b} \,, \\ \dfrac{(x - \overline{b})}{\theta \, \overline{b}} & \text{if } \overline{b} < x \,. \end{cases}$$

$$= \max \, (\underline{u} \; \underline{e}, \; 0, \; \overline{u} \; \overline{e}) \,.$$

This function is shown in Figure 2. Notice that the penalty is 0 for values of x within the range of permitted errors $(\underline{b}, \overline{b})$ and increases at a rate $1/(\underline{b} \; \psi)$ if x is less than $\underline{b}$ or $1/(\overline{b} \; \theta)$ if x is greater than $\overline{b}$. Also, notice that if $\underline{b} = b = \overline{b}$ then this objective is the same as the goal

TABLE 7

ERROR FOR ELASTIC PROGRAMMING

	by job type and rank	by people type and rank
shortfall	$\underline{e}_{ij} = \underline{b}_{ij} - x_{ij}$	$\underline{d}_{ik} = \underline{p}_{ik} - z_{ik}$
overfill	$\overline{e}_{ij} = x_{ij} - \overline{b}_{ij}$	$\overline{d}_{ik} = z_{ik} - \overline{p}_{ik}$

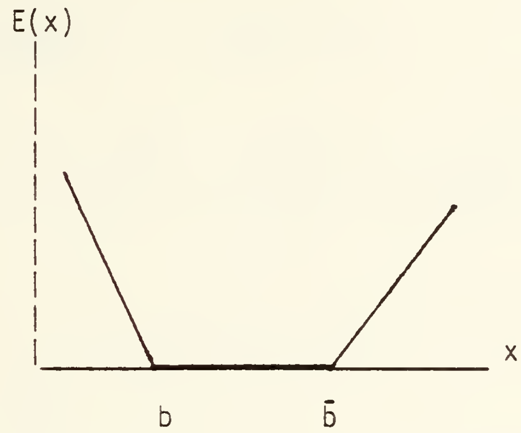

Fig. 2. Piecewise Function for Elastic Programming

programming objective. Thus, the goal program is a special case of the elastic program.

4. Quadratic Programming

Several quadratic programming formulations are presented in Section VI. These formulations are easy to use and give quick results for formulations that can be expressed without inequality constraints. The quadratic objective function to be used must have the same properties mentioned for the piecewise linear function and it must also be greater than zero. These properties may be stated mathematically as:

(i) $E(x) = 0$ if $x = b$,

(ii) $E(x) = 1$ if $x = \underline{b}$,

(iii) $E(x) = 1$ if $x = \overline{b}$.

(iv) $E(x) \geq 0$,

Unfortunately, quadratic functions which use the previously defined penalty measure must have symmetric penalties in order to satisfy the four conditions. That is, ψ must equal θ and δ must equal ϕ when considering the decision variable z. The following objective satisfies these conditions:

$$E(x) = \left(\frac{x - b}{\theta b}\right)^2 \qquad \text{when } \theta = \psi .$$

This type of penalty function is shown in Figure 3. An approximation that is useful if θ and ψ are close but not equal is described in Grinold [Ref. 1].

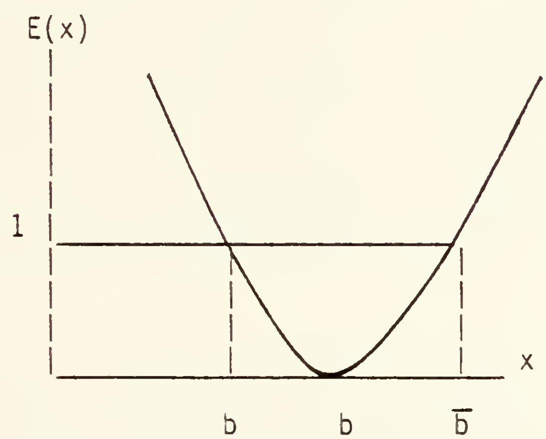

Fig. 3. Function for Quadratic Programming.

E. PROGRAMMING NOMENCLATURE

It will be useful to have a notational scheme to identify the various programming types, decision variables, and objective functions

found in the following sections. Let LP, EP, and QP refer to linear (goal), elastic, and quadratic programming, respectively. Let the objective function that sums errors by job class be a type 1 objective function, the objective function that sums the errors by people class be type 2, and the one that sums errors by both job class and people class be type 3. Thus, the mathematical program LP1(Y) is a linear program that uses the first objective function with Y as a decision variable. Similarly, QP2(Y,X) is a quadratic program with the second type of objective function that uses Y and X as the decision variables.

F. ADDING COST CONSIDERATIONS

Grinold [Ref. 1] also discusses a way in which cost considerations may be added to the model. Let c_k be the annualized cost of a type k accession. Then the multi-attribute objective function:

$$\lambda \left\{ \sum_k c_k y_k \right\} + (1 - \lambda) \left\{ \sum_i \sum_j E_{ij} + \sum_i \sum_k D_{ik} \right\}$$

where $0 \leq \lambda \leq 1$, will minimize the objective with respect to both cost and weighted penalties. This objective leads to a family of objective functions depending on the value of the parameter λ. Notice, however, that the units of the objective function are now in terms of cost units and percent error units.

The use of such multi-attribute functions is advantageous when the annualized cost of one type of manpower is significantly different than another type. For example, the annualized cost of a general unrestricted line officer is vastly less than that of a pilot due to training costs, pay bonuses, etc.

IV. LINEAR (GOAL) PROGRAMMING FORMULATION

A. GENERAL

A linear programming formulation for minimization of errors is found in Grinold [Ref. 1] and is derived from the goal programming concept as described in Lee [Ref. 5] and Hillier and Lieberman [Ref. 6]. Goal programming is a modification and extension of linear programming that allows a simultaneous solution of a system of complex objectives rather than a single objective. Notice that the goal of minimizing errors by differing ranks, job types, and people types constitutes a system of conflicting objectives. A solution which forces exact achievement of goals for the lower ranks may result in extreme deficits in the higher ranks while one which exactly meets requirements in the higher ranks may result in large excesses in the lower ranks. Thus, the goal approach is used to seek the best possible solution given the stated trade-offs.

In implementation of the following programs, the user should be aware of the tactical problems in solving LP's. It may be advantageous in some cases to solve the dual formulations of the problems depending on the number of variations and the number of constraints. An example of forming the dual for LP3(Y) and LP3(X,Y) is in Grinold [Ref. 1] and a general discussion of the topic is in Luenberger [Ref. 7].

B. ACCESSION AND INVENTORIES

The first formulation will be designated LP1(Y) and is shown in Figure 4. Recall that LP1 formulations involve errors of rank and

$$\min \sum_i \sum_j \left[u^{+}_{ij}\, e^{+}_{ij} + u^{-}_{ij}\, e^{-}_{ij} \right]$$

$$\text{s.t.} \quad \sum_k g_{kij}\, w_{ik}\, y_k + e^{-}_{ij} - e^{+}_{ij} = b_{ij} \qquad \text{for all } i,j$$

$$y_k,\ e^{-}_{ij},\ e^{+}_{ij} \geq 0 \qquad \text{for all } k,i,j$$

Fig. 4. LP1(Y)

job type. The constraint arises from the conservation of flow derived from equation (6) and sharing property in equation (7). An alternate formulation that considers both types of errors is LP3(Y) as shown in Figure 5. This is labelled in Grinold [Ref. 1] as LP-P.

$$\min \sum_i \sum_j \left[u^{+}_{ij}\, e^{+}_{ij} + u^{-}_{ij}\, e^{-}_{ij} \right] + \sum_i \sum_k \left[v^{+}_{ik}\, d^{+}_{ik} + v^{-}_{ik}\, d^{-}_{ik} \right]$$

$$\text{s.t.} \quad \sum_k g_{kij}\, w_{ik}\, y_k + e^{-}_{ij} - e^{+}_{ij} = b_{ij} \qquad \text{for all } i,j$$

$$w_{ik}\, y_k + d^{-}_{ik} - d^{+}_{ik} = p_{ik} \qquad \text{for all } i,k$$

$$y_k,\ e^{-}_{ij},\ e^{+}_{ij},\ d^{-}_{ik},\ d^{+}_{ik} \geq 0 \qquad \text{for all } k,i,j$$

Fig. 5. LP3(Y)

Figure 6 displays the formulation which uses both X and Y as the decision variables. It is LP3(X,Y) and is found in Grinold [Ref. 1] as LP-B.

$$\min \sum_i \sum_j \left[u^+_{ij} e^+_{ij} + u^-_{ij} e^-_{ij} \right] + \sum_i \sum_k \left[v^+_{ik} d^+_{ik} + v^-_{ik} d^-_{ik} \right]$$

$$\text{s.t. } x_{ij} + e^-_{ij} - e^+_{ij} = b_{ij} \quad \text{for all } i,j$$

$$w_{ik}\, y_k + d^-_{ik} - d^+_{ik} = p_{ik} \quad \text{for all } i,k$$

$$\sum_j f_{kij}\, x_{ij} - w_{ik}\, y_k = 0 \quad \text{for all } i,j$$

$$x_{ij},\ y_k,\ e^-_{ij},\ e^+_{ij},\ d^-_{ik},\ d^+_{ik} \geq 0 \quad \text{for all } k,i,j$$

Fig. 6. LP3(X,Y)

C. EXPECTED STAGE LENGTHS

Another application using goal programming would be to minimize the errors with respect to expected stage lengths (W). Recall that the stage length for rank i, person type k is the expected amount of time a recruit in class k will spend in rank i. An added constraint for this formulation is that the stage lengths have upper and lower bounds. In the Navy officer example, such bounds would reflect navy policy or legal restrictions imposed by Congress. Let these upper and lower bounds be defined by matrices $\underline{W}$ and $\overline{W}$, respectively. LP1(W) may be expressed as shown in Figure 7.

$$\min \sum_i \sum_j \left[u^+_{ij} \; e^+_{ij} + u^-_{ij} \; e^-_{ij} \right]$$

$$\text{s.t.} \quad \sum_k g_{kij} \; w_{ik} \; y_k + e^-_{ij} - e^+_{ij} = b_{ij} \quad \text{for all } i,j$$

$$e^-_{ij}, \; e^+_{ij} \geq 0 \quad \text{for all } k,i,j$$

$$\underline{W} \leq W \leq \overline{W}$$

Fig. 7. LP1(W).

D. SHARING RULES

Of particular interest is a mathematical program to specify the sharing fraction that represents the policy for distributing job classes among the people types (F) or people classes among job types (G). An additional constraint on this formulation is that F must sum to one over the index k and G must sum to one over the index j. Also, the sharing fractions may be bounded above and below. Using the same convention as used in the case of stage lengths, let $\underline{F}$ and $\underline{G}$ be the lower bounds and let $\overline{F}$ and $\overline{G}$ be the upper bounds on F and G, respectively. Notice that if no sharing is possible, specifying that both the upper and lower bound be zero sets the value at zero. LP1(G) and LP2(F) are formulated as shown in Figures 8 and 9.

$$\min \sum_i \sum_j \left[u^+_{ij} e^+_{ij} + u^-_{ij} e^-_{ij} \right]$$

$$\text{s.t.} \quad \sum_k z_{ik} g_{kij} + e^-_{ij} - e^+_{ij} = b_{ij} \quad \text{for all } i,j$$

$$\sum_j g_{kij} = 1 \qquad \text{for all } i,k$$

$$e^-_{ij}, e^+_{ij} \geq 0 \quad \text{for all } i,j$$

$$\underline{G} \leq G \leq \overline{G}$$

Fig. 8. LP1(G)

$$\min \sum_i \sum_k \left[v^+_{ik} d^+_{ik} + v^-_{ik} d^-_{ik} \right]$$

$$\text{s.t.} \quad \sum_j x_{ij} f_{kij} + d^-_{ik} - d^+_{ik} = p_{ik} \quad \text{for all } i,k$$

$$\sum_k f_{kij} = 1 \qquad \text{for all } i,j$$

$$e^-_{ij}, e^+_{ij} \geq 0 \quad \text{for all } i,j$$

$$\underline{F} \leq F \leq \overline{F}$$

Fig. 9. LP2(F)

E. INTERPRETATION FROM LP THEORY

In classical LP terms, the errors e^+, e^-, d^+, and d^- represent logical variables introduced to maintain the equality constraints. The errors e^- and d^- may be thought of as artificial variables and the errors e^+ and d^+ as surplus variables and hence, they measure the distance from a particular solution to the requirement. These objective functions are mathematically equivalent to a Lagrangian form with cost coefficients of the decision variables equal to zero (see Duff [Ref. 8: p. 64]). The weighting factors u^+, u^-, v^+, and v^- represent bounds on the variables of the dual formulation. In fact, an interpretation of a particular weighting factor is that it defines the penalty cost per unit violation of its associated constraint. Therefore, weighting factors which are approximately equal have associated constraints which are of the same relative importance, if binding. In other words, the incremental penalty for violating each such constraint is approximately the same. Such information should be of use in analyzing the results of these programs.

V. ELASTIC PROGRAMMING FORMULATION

A. GENERAL

The elastic programming model is described in Duff [Ref. 8]. The formulation is quite similar to that of the goal program in the last section. However, there is now a subtle but important difference. In the goal program, the objective was to allocate people as closely as possible to an exact number of billets (either B or P). In the elastic program, the objective is now to allocate people so as to be within a specified range of the billets to be filled. For this reason, the actual error will be indicated by a change in the notation to the use of underscore and overscore in place of superscripts plus and minus.

Before starting the formulations, it is necessary to define a variable to represent the slack from the upper bounds (B or P). Let q_{ij} and r_{ik} represent these slack variables for the (i,j) and (i,k) inventories. Also recall that $\underline{u}$, $\overline{u}$, $\underline{v}$, $\overline{v}$ are the weighting factors and $\underline{e}$, $\overline{e}$, $\underline{d}$, $\overline{d}$ are the errors from the desired ranges.

B. ACCESSION AND INVENTORIES

The first formulation shown in Figure 10 below is similar to the first formulation in the previous chapter and is denoted EP1(Y). The variable Y once again represents an upper bound on the accession vector Y. An alternate formulation that considers both type of errors is EP3(Y) presented in Figure 11.

$$\min \sum_i \sum_j \left[\underline{u}_{ij}\,\underline{e}_{ij} + \overline{u}_{ij}\,\overline{e}_{ij}\right]$$

$$\text{s.t.} \sum_k g_{kij}\, w_{ik}\, y_k + \underline{e}_{ij} - \overline{e}_{ij} + q_{ij} = \overline{b}_{ij} \quad \text{for all } i,j$$

$$0 \le Y \le \overline{Y}$$

$$0 \le Q \le \overline{B} - \underline{B}$$

$$\overline{e}_{ij},\ \underline{e}_{ij} \ge 0 \quad \text{for all } i,j$$

Fig. 10. EP1(Y)

$$\min \sum_i \sum_j \left[\underline{u}_{ij}\,\underline{e}_{ij} + \overline{u}_{ij}\,\overline{e}_{ij}\right] + \sum_i \sum_k \left[\underline{v}_{ik}\,\underline{d}_{ik} + \overline{v}_{ik}\,\overline{d}_{ik}\right]$$

$$\text{s.t.} \sum_k g_{kij}\, w_{ik}\, y_k + \underline{e}_{ij} - \overline{e}_{ij} + q_{ij} = \overline{b}_{ij} \quad \text{for all } i,j$$

$$w_{ik}\, y_k + \underline{d}_{ik} - \overline{d}_{ik} + r_{ik} = \overline{p}_{ik} \quad \text{for all } i,k$$

$$0 \le Y \le \overline{Y}$$

$$0 \le Q \le \overline{B} - \underline{B}$$

$$0 \le R \le \overline{P} - \underline{P}$$

$$\overline{d}_{ik},\ \underline{d}_{ik},\ \overline{e}_{ij},\ \underline{e}_{ij} \ge 0 \quad \text{for all } k,i,j$$

Fig. 11. EP3(Y)

C. EXPECTED STAGE LENGTHS AND SHARING RULES

Elastic programming may also be used to determine expected stage lengths or sharing rules. Only one example, EP2(F) in Figure 12, will be presented since the formulations are quite similar to those in the last section. Recall that an additional constraint exists for the sharing rules since they must sum to one over the appropriate index. An example of this is EP2(F) shown below.

$$\min \sum_i \sum_k \left[\underline{v}_{ik}\,\underline{d}_{ik} + \overline{v}_{ik}\,\overline{d}_{ik}\right]$$

$$\text{s.t.} \quad \sum_j x_{ij}\, f_{kij} + \underline{d}_{ik} - \overline{d}_{ik} + r_{ik} = \overline{p}_{ik} \quad \text{for all } i,k$$

$$\sum_k f_{kij} = 1 \quad \text{for all } i,j$$

$$\underline{F} \le F \le \overline{F}$$

$$0 \le R \le \overline{F} - \underline{P}$$

$$\overline{d}_{ik},\ \underline{d}_{ik} \ge 0 \quad \text{for all } i,k$$

Fig. 12. EP2(F)

D. INTERPRETATION FROM LP THEORY

As was the case for the linear (goal) program, the elastic program is mathematically equivalent to a Lagrangian form. The variables $\underline{e}$ and $\underline{d}$ are artificial and measure the distance from a particular solution to the lower error bound. The surplus variables $\overline{e}$ and $\overline{d}$ measure the distance above the upper error bound and the slack variables q and r measure the

distance from a solution to the associated upper bound. The weighting factors continue to represent bounds on the dual formulation.

This formulation has an advantage in that the solution space for zero penalty is greatly expanded even if the error range is small. This has improtant consequences for the implementation and achievement of acceptable solutions for problems with many constraints and variables.

VI. QUADRATIC PROGRAMMING FORMULATIONS

A. GENERAL

The quadratic programming model described in Grinold [Ref. 1] provides an alternate method for optimization of allocations. A general discussion of the solution techniques for quadratic programs is found in Simmons [Ref. 9]. The following programs are expressed with only equality constraints and will yield solutions that are found by analytically solving systems of equations. This provides a useful and consistent means of determining allocations. However, the use of QP as presented here must be weighed against two disadvantages.

- The penalty weights must be symmetric for both deficit and excess types of errors.
- The decision variables are unbounded and, thus, the QP may yield impractical solutions such as negative accessions.

Since much of the data input for the model is subjective or highly variable, and the purpose of the model is to explore questions concerning policy planning, the magnitudes and relative relationships of the allocation are more important than an exact choice of allocation. Thus, the use of symmetric penalties may not present significant problems. If the parameters and data inputs to the model are fairly consistent with real world conditions, then the use of unbounded variables is unlikely to produce inconsistent solutions which would adversely affect the result of the analysis. The appearance of negative values in the solution is likely to indicate misspecification of parameters or the attempt to

achieve unrealistic goals. For the following sections, it will be assumed that $\psi = 0$ and $\delta = \phi$.

The QP is similar to a least squares approach in that the purpose is to minimize the sum of squared weighted errors. Appendix B contains a brief summary of the resulting quadratic forms used in this section. Reindexing the variables makes it possible to conveniently express the QP as a system of equations using matrix arithmetic notation. For mathematical convenience, the following change of variables may be necessary:

$$m = J\,(i - 1) + j \tag{8}$$

where $m=1,2,\ldots M$ and $M=IJ$;

$$n = K\,(i - 1) + k \tag{9}$$

where $n=1,2,\ldots,N$ and $N=IK$.

In this way, matrices of size (I,K) may be reindexed to become vectors of size (N) and matrices of size (I,J) become vectors of size M. Three dimensional arrays may also be reduced to two dimensions. For example, a three dimensional array of size (I,J,K) indexed over (i,j,k) may be reindexed as a two dimensional array of size (M,K) using equation (8). In using matrix notation, it will be necessary on occasion to reform a vector into a diagonal matrix. This means that the k^{th} element of the vector will occupy the $(k,k)^{th}$ position of the matrix and all off diagonal elements will be zero.

B. ACCESSION AND INVENTORIES

The first formulation to be considered is QP1(Y). The advantage of this formulation is that it requires less data input since the job sharing fractions (F) and requirements by rank and manpower class (P) are not needed. Also, once the optimal accession and inventories are

determined, values for F and P may be computed based on the optimization results. The objective function is:

$$\min \sum_i \sum_j u_{ij}^2 \left(\sum_k g_{kij}\, w_{ik}\, y_k - b_{ij} \right)^2 .$$

Now let $h_{ijk} = g_{kij}\, w_{ik}$ and then reindex (i,j) to (m) and (i,j,k) to (m,k) as given above in equation (8). This results in:

$$\min \sum_m u_m^2 \left(\sum_k h_{mk}\, y_k - b_m \right)^2$$

which, in matrix notation, equals:

$$\min\ (HY - B)'\, U'U\, (HY - B)$$

where U is now an (M,M) diagonal matrix of the penalty weights. Taking the derivative with respect to Y and setting the result equal to zero yields the following solution:

$$Y = (H'U'UH)^{-1}\, H'U'UB .$$

Once an optimal value for Y has been determined, inventories may be calculated using equations (7) and (6) from Section II.

The formulation QP3(Y) minimizes the objective:

$$\min \sum_i \sum_j u_{ij}^2 (x_{ij} - b_{ij})^2 + \sum_i \sum_k v_{ik}^2 (w_{ik}\, y_k - p_{ik})^2$$

subject to the sharing constraint

$$x_{ij} = \sum_k g_{kij}\, w_{ik}\, y_k .$$

Substituting the constraint into the objective, letting $h_{ijk} = g_{kij}\, w_{ik}$, and then reindexing from (i,j,k) to (m,k) as given above in equation (8), yields in matrix notation:

$$\min\ (HY - B)'\, U'U\, (HY - B) + Y'QY - 2Y'R + c$$

where Q is a (K,K) diagonal matrix whose diagonal elements are given by

$$q_{kk} = \sum_i v_{ik}^2\, w_{ik}^2 ; \qquad (10)$$

R is a (K) vector with elements $r_k = \sum_i v_{ik}^2 p_{ik} w_{ik}$;

and c is a scalar constant, namely $c = \sum_i \sum_k v_{ik}^2 p_{ik}^2$.

Taking the derivative and setting it equal to zero results in

$$H'U'UHY - H'U'UB + QY - R = 0 .$$

Finally, the solution is:

$$Y = [H'U'UH + Q]^{-1} [H'U'UB + R] .$$

This model can be found in Grinold [Ref. 1] as UQ-P. An advantage to the use of this model over other QP3 formulations is that it uses the people sharing rule G which may be more available as a data input than the job sharing rule F.

An alternative program which uses the same objective function is QP3(X,Y). In this case, minimize the objective subject to the sharing constraint:

$$w_{ik} y_k - \sum_j f_{kij} x_{ij} = 0 \text{ for all } i,k.$$

Both X and Y are now the decision variables and the constraint cannot be substituted into the objective function directly. In order to express the constraint in matrix notation, it is necessary to define two new matrices, F* and W*. First, F* is defined by the elements:

$$f^*_{i'kij} = \begin{cases} f_{kij} & \text{if } i' = i \\ 0 & \text{otherwise} \end{cases}$$

where $i, i'=1,\ldots,I,\ k=1,\ldots,K,$ and $j=1,\ldots,J.$

Now reindex from (i',k,i,j) to (n,m) as given by equations (8) and (9) above to form the properly shaped F*. This results in a block matrix in which each of the blocks is a diagonal matrix of the j^{th} column of F for a particular (i,k). W* is formed in a similar manner. Let:

$$w^*_{ik'k} = \begin{cases} w_{ik} & \text{for } k = k' \\ 0 & \text{otherwise} \end{cases}$$

where $k, k'=1,\ldots,K$ and $i=1,\ldots,I$.

Next reindex W* from (i,k',k) to (n,k) according to (9) above.

The result is a diagonal block matrix in which each of the blocks on the diagonal is a column of W. Finally, the constraint can be expressed in matrix notation as:

$$F^*X - W^*Y = 0 \ .$$

The objective function in matrix notation is:

$$\min\ (X - B)'\ U'U\ (X - B) + Y'QY - 2Y'R + c$$

where U, Q, R, and c are the same as defined for QP3(Y). Now form the Lagrangian:

$$(X - B)'\ U'U\ (X - B) + Y'QY - 2Y'R + c - \lambda\ (F^*X - W^*Y)$$

where λ is the vector of Lagrange multipliers. Taking the derivative with respect to X, Y, and λ yields the following system of equations:

$$\begin{aligned} 2U'UX - 2U'UB - F^{*\prime} &= 0, \\ 2QY - 2R + W^* &= 0, \\ F^*X - W^*Y &= 0. \end{aligned}$$

This can be written in partitioned matrix form as:

$$\left[\begin{array}{c|c|c} U'U & 0 & -(1/2)F^{*\prime} \\ \hline 0 & Q & (1/2)W^* \\ \hline F^* & -W^* & 0 \end{array}\right] \left[\begin{array}{c} X \\ \hline Y \\ \hline \lambda \end{array}\right] = \left[\begin{array}{c} U'UB \\ \hline R \\ \hline 0 \end{array}\right]$$

where 0 indicates a zero matrix of appropriate size. The optimal Y, X, and λ is given by:

$$\begin{bmatrix} X \\ \hline Y \\ \hline \lambda \end{bmatrix} = \left[\begin{array}{c|c|c} U'U & 0 & -(1/2)F^{*\prime} \\ \hline 0 & Q & (1/2)W^* \\ \hline F^* & -W^* & 0 \end{array}\right]^{-1} \begin{bmatrix} U'UB \\ \hline R \\ \hline 0 \end{bmatrix}$$

The program QP3(X,Y) is found in Grinold [Ref. 1] with the name UQ-B. As can be seen from the result, QP3(X,Y) has a more complicated solution than QP3(Y) and has an additional disadvantage in that the results may slightly violate the consistency rule. Thus, a determination must be made whether to use both X and Y from the results or to construct the inventories solely on the basis of Y. It has been presented here as an illustration of the use of Lagrange multipliers and as a means of checking the results from QP3(Y).

Another means of checking QP3(Y) which insures that the consistency rule is met would be to solve QP3(X), use equation (5) to determine z, and then use the least squares approach presented in Section II to find values for Y. Those values of Y may then be used to reconstruct the inventories and insure consistency.

C. EXPECTED STAGE LENGTHS

The next formulation that will be discussed is QP2(W). The objective function is:

$$\min_{w} \sum_{i}\sum_{k} v_{ik}^2 \, (y_k \, w_{ik} - p_{ik})^2$$

Let $q_{ik} = v_{ik}^2 \, y_k$ and $r_{ik} = v_{ik}^2 \, p_{ik}$.

Then the result is:

$$\min_{w} \sum_{i}\sum_{k} (q_{ik} \, w_{ik} - r_{ik})^2 .$$

Now reindexing from (i,k) to (n) and changing to matrix notation yields:

$$\min \, (QW - R)' \, (QW - R)$$

where Q is reformed to be an (n,n) diagonal matrix. The solution to this program is:

$$W = [Q'Q]^{-1} \, Q'R$$

D. ADDING ADDITIONAL EQUALITY CONSTRAINTS

The technique of Lagrange multipliers presented with the QP3(X,Y) model may also be used with additional equality constraints. For example, suppose the total number of accessions to be brought into the system is known but the exact number in each manpower class is not known. This information may be added as an additional equality constraint:

$$\sum_{k} y_k = r$$

where r is now the total number of recruits to be brought into the system each time period. In matrix notation, this constraint becomes:

$$\varepsilon' Y - r = 0$$

where ε is a vector of k ones. Using the QP1(Y) model as an example, form the Lagrangian with this additional constraint to get

$$\min \, (HY - B)' \, U'U \, (HY - B) - \lambda \, (\varepsilon' Y - r) \; .$$

Take the derivatives, set equal to zero, and solve as before to get the solution:

$$\begin{bmatrix} Y \\ \hline \lambda \end{bmatrix} = \left[\begin{array}{c|c} H'U'UH & -(1/2)1 \\ \hline \varepsilon' & -r \end{array}\right]^{-1} \begin{bmatrix} H'U'UB \\ \hline 0 \end{bmatrix}$$

The advantage of adding an additional constraint is that it can be used to force the quadratic model to search for solutions that will be more related to "real world" problems. This helps to mitigate the disadvantage regarding unbounded decision variables. In fact, a range of possible values could be specified and then the optimal solution within the range could be found using a simple numerical search procedure. Since the solution to each problem during a search iteration is explicit and the resolution needed is to the nearest whole integer, such a procedure is easy to implement and should arrive at solutions fairly quickly.

VII. DEMONSTRATION OF THE MODEL

A. IMPLEMENTATION

The model was implemented in its quadratic form on an IBM 3033 computer at the Naval Postgraduate School using the APL programming language as shown in Appendix C. For the linear and elastic programs, preprocessors were written in APL to arrange the data in standard MPS programming format for implementation on the XS-system optimization package. MPS is an international standard and such format is easily transferrable to other types of computers. A sample output is included in the program listing section. In addition, listings of some of the APL program functions used to produce the results in this section are also included. The ease with which data arrays can be manipulated in APL resulted in concise and flexible program functions which performed the optimizations and the input-output formatting of the data. Also written were two "user friendly" functions for data input and data display, and a master driver function that allows the user to select from a menu of options. Appendix C contains a brief summary of these programs.

B. SYSTEM DESCRIPTION

1. Attributes

This demonstration is based on an example from the U.S. Navy Officer Corps. The attributes chosen for manpower classes, career stages, and job classes are the same as those found in Grinold [Ref. 1] and are shown in Figure 13.

MANPOWER CLASSES

Type	Officer Type	Description
1	110x	General Unrestricted Line (GURL)
2	111x/116x	Surface Warfare
3	112x/117x	Submarine Warfare
4	131x/139x	Pilots
5	132x/137x	Naval Flight Officer (NFO)

CAREER STAGE

Stage	Rank	Years of Service
1	ENS	0 - 2
2	LTJG	2 - 4
3	LT	4 - 9
4	LCDR	9 - 14
5	CDR	14 - 19
6	CAPT	19 - 26

JOB CLASSES

Type	Billet Designator	Description
1	1000	General, Nonwarfare
2	1050	General, Warfare
3	1110/1160	Surface Warfare
4	1120/1170	Subsurface Warfare
5	1310/1390	Pilots
6	1320/1370	Naval Flight Officer
7	1300	General Aviation

Fig. 13. Attributes for Example.

2. Expected Time in Each Rank

To compute W, it is necessary to have stage lengths and survivor fractions. Navy usage of survivor fractions is usually found in the form of continuation or retention rates. Current data was obtained from Deputy Chief of Naval Operations (MPT), OP-01, with continuation rates for Navy Unrestricted Line. The continuation rates were used directly to compute expected length of time spent in each rank. This data is shown in Appendix D. The APL function WAITS uses continuation rates and stage

lengths as inputs and produces the matrix W. the program GWAITS will also accomplish this but uses survivor fractions instead of retention rates. The resulting steady state matrix W is shown in Table 8.

TABLE 8

EXPECTED STAGE LENGTHS

	GURL	SURF	SUB	PILOT	NFO
ENS	1.925	1.942	1.891	2.000	1.992
LTJG	1.674	1.700	1.621	1.975	1.887
LT	2.063	2.073	2.040	2.827	3.241
LCDR	1.106	1.138	1.040	1.028	2.255
CDR	0.856	0.896	0.777	0.773	1.869
CAPT	0.596	0.589	0.607	0.445	1.276

3. Billet Requirements and Sharing Rules

Information concerning the requirements by rank and job type was also obtained from OP-01 and the array B was constructed. (See Table 9). The next parameter to be specified is the billet sharing array. This information was taken from Grinold [Ref. 1] with some slight modifications and is displayed in Table 10. Once B and F are available, it is possible

TABLE 9

BILLET REQUIREMENTS BY RANK AND JOB TYPE

	1000	1050	1110	1120	1310	1320	1300
ENS	330	0	2258	660	1105	593	0
LTJG	768	0	1706	700	1970	1185	2
LT	1780	378	2145	884	3780	1408	629
LCDR	1456	571	1510	877	1818	586	716
CDR	1001	468	990	518	710	54	909
CAPT	571	462	354	152	0	0	422

TABLE 10

BILLET SHARING ARRAY

GURL	1000	1050	1110	1120	1310	1320	1300
ENS	0.73	0	0	0	0	0	0
LTJG	0.70	0	0	0	0	0	0
LT	0.70	0	0	0	0	0	0
LCDR	0.70	0	0	0	0	0	0
CDR	0.70	0	0	0	0	0	0
CAPT	0.70	0	0	0	0	0	0

SURF	1000	1050	1110	1120	1310	1320	1300
ENS	0.18	1.00	1	0	0	0	0
LTJG	0.15	0.50	1	0	0	0	0
LT	0.15	0.50	1	0	0	0	0
LCDR	0.15	0.50	1	0	0	0	0
CDR	0.15	0.50	1	0	0	0	0
CAPT	0.15	0.50	1	0	0	0	0

SUB	1000	1050	1110	1120	1310	1320	1300
ENS	0.09	0.00	0	1	0	0	0
LTJG	0.06	0.10	0	1	0	0	0
LT	0.06	0.20	0	1	0	0	0
LCDR	0.06	0.20	0	1	0	0	0
CDR	0.06	0.20	0	1	0	0	0
CAPT	0.06	0.20	0	1	0	0	0

PILOT	1000	1050	1110	1120	1310	1320	1300
ENS	0.00	0.00	0	0	1	0	0.00
LTJG	0.50	0.10	0	0	1	0	0.00
LT	0.50	0.18	0	0	1	0	0.57
LCDR	0.50	0.18	0	0	1	0	0.57
CDR	0.50	0.18	0	0	1	0	0.57
CAPT	0.50	0.18	0	0	1	0	0.57

NFO	1000	1050	1110	1120	1310	1320	1300
ENS	0.00	0.00	0	0	0	1	1.00
LTJG	0.40	0.10	0	0	0	1	1.00
LT	0.40	0.12	0	0	0	1	0.43
LCDR	0.40	0.12	0	0	0	1	0.43
CDR	0.40	0.12	0	0	0	1	0.43
CAPT	0.40	0.12	0	0	0	1	0.43

to calculate the corresponding requirements in terms of people types and ranks, and the corresponding people-sharing array. The results for the requirements by people types is shown in Table 11. An example of the people-sharing array is shown in Table 12 for the manpower class of pilots.

TABLE 11

BILLET REQUIREMENTS BY RANK AND MANPOWER CLASS

	GURL	SURF	SUB	PILOT	NFO
ENS	241	2317	690	1105	593
LTJG	538	1821	746	2010	1217
LT	1246	2601	1066	4296	1795
LCDR	1019	2014	1079	2402	1021
CDR	701	1374	672	1362	541
CAPT	400	671	279	352	260

TABLE 12

PEOPLE-SHARING ARRAY FOR PILOTS

	1000	1050	1110	1120	1310	1320	1300
ENS	0.015	0.000	0	0	0.985	0	0.000
LTJG	0.019	0.000	0	0	0.980	0	0.001
LT	0.021	0.016	0	0	0.880	0	0.083
LCDR	0.030	0.043	0	0	0.757	0	0.170
CDR	0.037	0.062	0	0	0.521	0	0.380
CAPT	0.081	0.236	0	0	0.000	0	0.683

4. Permitted Unit Errors

The permitted unit errors were determined based on subjective judgment. First the matrix for permitted unit errors by rank and job type was estimated. This is shown in Table 13. The entries marked as 1

TABLE 13

PERMITTED UNIT ERRORS BY JOB TYPE AND RANK

	1000	1050	1110	1120	1310	1320	1300
ENS	400	1	500	500	500	400	1
LTJG	400	1	500	500	500	400	500
LT	10	8	8	10	8	10	10
LCDR	10	8	10	15	8	8	10
CDR	10	8	5	10	8	10	10
CAPT	10	10	10	10	1	1	10

correspond to ranks and job types for which there are no billets. Giving a very low percentage will help ensure the optimization does not seek to place people in these jobs. The corresponding errors by people type and rank were generated by the use of the consistency rule. The unit errors do not necessarily have to meet this requirement, but it was felt that the consistency rule would give an approximation as reasonable as further use of subjective judgment. The results were checked to insure that the numbers were reasonable, and no changes were made. The matrix for permitted unit errors by rank and people type is shown in Table 14. These last two matrices represent the permitted error for overfilling the billets. The matrices for underfilling the billets were done using a similar method.

C. ACCESSIONS USING QUADRATIC PROGRAMMING

The first set of programs to be considered are those labelled QP1(Y), QP2(Y), and QP3(X,Y). The results of the optimization with respect to the accessions are shown in Table 15. The column marked "Obj. Val." is the value of the penalty function that measures both types of errors. Notice that QP1(Y) which only minimized with respect to errors by job

TABLE 14

PERMITTED UNIT ERRORS BY RANK AND PEOPLE TYPE

	GURL	SURF	SUB	PILOT	NFO
ENS	280.0	560.5	524.2	520.7	416.5
LTJG	280.0	560.5	524.2	805.2	631.1
LT	7.0	13.5	12.2	15.6	15.7
LCDR	7.0	15.5	17.2	15.6	13.7
CDR	7.0	10.5	12.2	15.6	15.7
CAPT	7.0	16.5	12.6	9.0	6.9

TABLE 15

ACCESSION USING QUADRATIC PROGRAMMING

	Obj. Val.	GURL	SURF	SUB	PILOT	NFO	Total
QP1(Y)	455.2	704	1404	554	1470	332	4464
QP3(Y)	390.6	712	1416	563	1246	272	4209
QP3(Y,X)	400.8	704	1380	556	1388	270	4298

type and rank still did well when compared to the other two. In fact, as can be seen from the actual accession values, all three programs had remarkably close results. Since QP3(Y) did the best in terms of the value of the objective function, the results from this optimization will be presented in the remainder of this section.

While close, the results from all three programs have rather high penalty values. Recall that the minimum value that can be achieved is zero, and the actual magnitude of the penalty values shows that our actual allocation is far off the mark. Another way to gain an appreciation of how close the allocation is to the targets is by examining the percent error compared to the permitted errors. Listed below in Table 16

TABLE 16

PERCENT ERROR BY RANK AND JOB TYPE

	1000	1050	1110	1120	1310	1320	1300
ENS	351	0	19	54	125	-9	0
LTJG	90	0	32	22	22	-58	-12
LT	12	-1	13	8	-18	-51	-32
LCDR	-26	-32	-20	-46	-47	-40	-44
CDR	-14	-17	-8	-35	-29	-6	-19
CAPT	14	31	24	23	0	0	47

are the percent errors from the QP3(Y) program. This table highlights a problem that is probably due to the high rate of attrition of the middle level officers. Notice that those ranks are underfilled (as indicated by the minus sign) while the junior and senior ranks are experiencing an overfill. The most serious problem exists in the rank of Lieutenant Commander and in the job classes Submarine Warfare Officer and Pilot. The model could be made to actually fill those shortage billets by moving the permitted unit error in those ranks and job classes closer to zero. However, since the current parameters are already closer to zero than any of the others, such changes would only result in worse overfills at the high and low end. A better approach would be to revise the sharing rules in an attempt to get a better allocation.

VIII. SUMMARY

A. GENERAL

The purpose of this thesis was to investigate and enhance the model proposed by Grinold [Ref. 1] with special attention to the optimization techniques and considerations involved with implementation. The model looks at requirements and inventories of a manpower system by classifying them according to people type, rank, and job type. By the use of sharing fractions, which represent management policy, it is possible to express billets which are usually defined by job type as requirements for certain types of people. The same sharing rules also make it possible to express inventories of people by the types of jobs filled rather than by the types of people filling those jobs. Further, the assumption of steady state makes it possible to calculate all inventories from accessions. Thus, billets, inventories, and accessions are linked by a unified and rather elegant model structure.

If relative trade-offs between different billets in the form of permitted shortfalls or overfills can be made, then it is possible to measure the error between the desired billet structure and the steady state inventories. Various optimization techniques may then be used to examine the long term result of policy decisions regarding the allocation of people to jobs and to test alternative policies.

This thesis has included several areas of research not presently found in Grinold [Ref. 1]. Several considerations were discussed concerning the implementation of the model. These include: level of detail in

attribute definitions, problems encountered in implementation, acquisition of model parameters, intuitive explanation of weighting factors, and advantages of various optimization schemes. New programming formulations were presented that used accessions, inventories, sharing rules, or expected stage lengths as the decision variables. Elastic programming was introduced as a generalization of the goal programming technique. Explicit solutions were derived for the quadratic programming model and the idea of additional equality constraints was proposed. Other enhancement features included the least squares approach to determine the requirements by people type and the job sharing rules given the billet structure and the people sharing rules, and using least squares to determine accessions from a given inventory.

In conclusion, this model allows the policy planner to examine a wide range of policy options using only a few simplifying assumptions and a modest amount of input data. Many areas of policy planning may be illuminated by simply placing a problem in the framework of the model since this requires that the analyst consider trade-offs between various people and job types and formalize policy in setting the values for the sharing fractions. This ability to examine the aggregate effect of human behavior on an organization's billet structure makes this model a useful tool for the policy planner.

B. AREAS FOR FURTHER STUDY

There are several areas related to this model where further research should be done. Suggested areas include the following:

- Extend the model structure to handle dynamic conditions. If the current inventory is known and the continuation rates of

personnel can be estimated, then a year-by-year approach could be used to examine various short-range policy problems. This would eliminate the limitation of steady state assumptions in the current model.

- The optimizations rely heavily on reasonable estimates of the permitted percent error in failing to fill all billets. More research is needed on how to gather, scale, and interpret data that could be used by the model with regard to these permitted errors.
- More improvement is needed in adding cost considerations to the model. The ability to relate any analysis to "the bottom line" should enhance the results of such an analysis. The idea of a multi-attribute objective function that considers both errors and costs was presented in Section III. Another possibility would be to develop a means to econometrically express the trade-off parameters.
- The usefulness of the model would be extended if the results could be expressed in such a manner that the impact of policy decisions could be quickly evaluated. One idea is to use the model to create ratios or indices that pinpoint the criticality of various allocations. For example, if the "optimal" billet structure based on projected inventories were calculated, then the ratio of this theoretical structure and the actual structure should provide a measure of "how critical" certain billets will be in the future. Ratios close to one are desirable, and those far away highlight the problem areas.

- Since the model is based on steady state assumptions, it is desirable to be able to relate the results of the model to these assumptions. Assuming certain functions that describe the way in which people leave the organization could be used to establish such a relationship. The impact of changes in continuation rates or survivor fractions on the billet structure needs to be directly related to the sharing policies that will be required to reduce the allocation errors, and conversely, changes in the billet structure need to be analyzed in light of the behavior of people in the organization.
- Further model enhancement is needed in order to introduce the effects of stage substitution to the model. In military organizations, such stage substitutions exist in the form of "selected but not yet promoted." The model has assumed that such stage substitution (i.e., personnel of one rank filling a job of a higher rank) is negligible or has a net effect of zero. The ability to model this would add more credibility to the model and subsequent analysis.

APPENDIX A

VARIABLE LIST FOR THE MODEL AND PROGRAMS

Name	Size	Description
k		The index of the manpower classes. $k=1,2,\ldots,K$.
i		The index of ranks or career stages. $i=1,2,\ldots,I$.
j		The index of the job classes. $j=1,2,\ldots,J$.
B	I,J	The number of people in rank i required to fill class j jobs.
P	I,K	The number of required people in rank i and manpower class k.
T	K,I,J	The number of (i,k) people required in type (i,j) jobs.
X	I,J	The inventory of people in rank i who are filling job j.
Z	I,K	The inventory of people in rank i and manpower class k.
A	K,I,J	The inventory of (i,k) people who fill type j jobs.
F	K,I,J	The fraction of (i,j) jobs that should be performed by manpower class k.
G	K,I,J	The fraction of type (i,k) manpower who should fill job type j.
W	I,K	The expected length of time a person in manpower class k will spend in rank i.
S	I,K	The fraction of type k people who will remain in the organization past rank i.
ψ	I,J	The permitted unit error for underfilling job class j and rank i.
θ	I,J	The permitted unit error for overfilling job class j and rank i.
δ	I,K	The permitted unit error for underfilling rank i and manpower class k.
ϕ	I,K	The permitted unit error for overfilling the rank i and manpower class k.
$\underline{B}$	I,J	The lower bound of permitted error for job class j and rank i.

$\overline{B}$	I,J	The upper bound of permitted error for job class j and rank i.
$\underline{P}$	I,K	The lower bound of permitted error for manpower class k and rank i.
$\overline{P}$	I,K	The upper bound of permitted error for manpower class k and rank i.
C	K,J	An indicator that is 1 if manpower class k can fill job j; 0 otherwise.
e	I,J	The error in meeting $(i,j)^{th}$ requirement.
d	I,K	The error in meeting $(i,k)^{th}$ requirement.
u	I,J	The weighting factor for an (i,j) error.
v	I,K	The weighting factor for an (i,k) error.

APPENDIX B

QUADRATIC FORMS FOR THE MODEL

1. The General Form

The quadratic objective functions in this thesis have the same general quadratic form which can be expressed in summation notation or matrix notation, if properly reindexed. Consider the quadratic form with variables X(I,1), A(I,1), P(I,1) and constant r:

$$\sum_i a_i^2 x_i^2 - 2 \sum_i p_i x_i + r \ . \qquad \text{(a1)}$$

In matrix form this becomes:

$$X'QX - 2X'P + r \qquad \text{(a2)}$$

where Q is the diagonal matrix:

$$Q = \begin{bmatrix} a_1^2 & & \\ & \ddots & \\ & & a_I^2 \end{bmatrix}$$

2. The Special Form

If the above general quadratic form has the following characteristics, then a more familiar form can be written for it.

$$\text{Let } r = \sum_i b_i^2 \quad \text{and} \quad p_i = a_i b_i$$

and then (a1) becomes:

$$\sum_i a_i^2 x_i - 2\sum_i a_i b_i x_i + \sum_i b_i^2 \ ,$$

$$\sum_i (a_i x_i - b_i)^2 \ . \qquad \text{(a3)}$$

In matrix notation this is:

$$X'D'DX - 2X'D'B + B'B$$

$$(DX - B)' \ (DX - B) \tag{a4}$$

where D is the diagonal matrix:

$$D = \begin{bmatrix} a_1 & & \\ & \ddots & \\ & & a_I \end{bmatrix}$$

This special form has several cases which are of interest.

Case 1

Let $a_i = 1$. Then (a3) becomes:

$$\sum_i (x_i - b_i)^2 \tag{a5}$$

or in matrix form:

$$(X - B)' \ (X - B) \tag{a6}$$

Case 2

Let $b_i = a_i c_i$. Then (a3) becomes:

$$\sum_i a^2 \ (x_i - c_i)^2 \tag{a7}$$

$$X'D'DX - 2X'D'DC - C'D'DC$$

$$(DX - DB)' \ (DX - DB)$$

$$(X - C)' \ D'D \ (X - C) \quad . \tag{a8}$$

Case 3

Let $b_i = a_i c_i$ and $x_i = \sum_i f_{ik} z_k$.

Since FZ = X, this form is like (a7) and has similar results:

$$\sum_i a_i^2 \ [(\sum_k f_{ik} z_k) - c_i]^2 \tag{a9}$$

or in matrix notation:

$$(FZ - C)' \ D'D \ (FZ - C) \quad . \tag{a10}$$

APPENDIX C

APL FUNCTIONS

Function	Uses	Computes	Syntax and Description
QUAD	All variables		QUAD. Master Driver Program. This function calls all others through the use of menu options.
QP3XY	B, F, P, W, θ, ϕ	Y, X, Z, A	QP3XY. Calculates "optimal" accessions Y, personnel inventory Z, and allocation A, given the data, objective, and the billet share rule F.
MAKEPG	P, G	B, F, T	MAKEPG. Reconciles the global variables B, F, and T with the global variables P and G.
MAKEBF	B, F	P, G, T	MAKEBF. Reconciles the global variables P, G, and T with the global variables B and F.
MAKEFP	B, G	P, F, T	MAKEFP. Reconciles the global variables P, F, and T with the global variables B and G using least squares approach.
QP3Y	G, B, F, P, W, θ, ϕ	Y, X, Z, A	QP3Y. Calculates "optimal" accessions Y, personnel inventory Z, billet staffing X, and allocation A, given the data, and use of the people share rule.
INVENTORY	G, Y, W	Z, X, A	INVENTORY. Calculates inventory Z, X and allocation A, where accessions, people share rule, and expected stage lengths are given.
PEOPRCNT	P, Z	100(D-Z)/P	PEOPRCNT. Calculates the percentage error in inventory. Compare with ϕ.

Function	Uses	Computes	Syntax and Description
BILPRCNT	B, X	100(B - x)/B	BILPRCNT. Calculates the percentage error in meeting billet requirements. compare with θ.
GWAITS	ALPHA, S	W	GWAITS. Given the survivor fractions ALPHA and stage definitions S, calculates W the expected waiting time in each stage.
WAITS	RET, S	W	WAITS. Given the retention rates RET and stage definitions S, calculates W, the expected waiting time in each stage.
ROUND			ROUND XXX. Takes any array and rounds elements to integers.
LABIJ	ILAB, JLAB, DP, CS		LABIJ XXX. Takes any array of dimension (I,J) and labels rows and columns according to ILAB an JLAB. ILAB must have I rows and JLAB must have J rows. DP specifies number of decimal places desired. CS specifies minimum column spacing.
LABIK	ILAB, KLAB, DP, CS		LABIK XXX. Handles any array of dimension (I,K) in the same manner as LABIJ.
LABKIJ	ILAB, KLAB, JLAB, DP, CS		LABKIJ XXX. Handles any array of dimension (K, I, J) in the same manner as LABIJ.
DISPIN	G, F, B, P, ILAB, KLAB, JLAB, DP, CS		DISPIN. User selects from a menu to produce the input variable with appropriate labels.
DISPOUT	X, Y, Z, B, ILABS, KLAB, JLAB, DP, CS		DISPOUT User selects from a menu to produce the output variable with appropriate labels. Also displays error arrays from BILPRONT and PEOPRCNT.

APPENDIX D

NAVY URL CONTINUATION RATES [1]

Years	GURL	SURF	SUB	PILOT	NFO
1	0.963	0.971	0.947	1.000	1.000
2	0.960	0.971	0.938	1.000	0.985
3	0.930	0.928	0.933	0.990	0.970
4	0.820	0.809	0.841	0.980	0.919
5	0.698	0.698	0.697	0.882	0.858
6	0.816	0.819	0.810	0.692	0.882
7	0.870	0.860	0.891	0.706	0.859
8	0.912	0.914	0.908	0.808	0.959
9	0.947	0.970	0.902	0.818	0.972
10	0.843	0.833	0.862	0.828	0.934
11	0.868	0.884	0.835	0.883	0.918
12	0.922	0.922	0.922	0.933	0.927
13	0.914	0.926	0.890	0.969	0.971
14	0.981	0.987	0.970	0.964	0.986
15	0.974	0.971	0.979	0.954	0.974
16	0.989	0.990	0.987	0.948	0.974
17	0.978	0.974	0.987	0.957	0.970
18	0.956	0.964	0.941	0.950	0.936
19	0.917	0.901	0.948	0.838	0.954
20	0.762	0.754	0.778	0.765	0.789
21	0.831	0.821	0.851	0.774	0.692
22	0.908	0.883	0.957	0.901	0.914
23	0.875	0.853	0.920	0.844	0.860
24	0.911	0.894	0.946	0.852	1.000
25	0.848	0.847	0.849	0.838	1.000
26	0.701	0.710	0.682	0.778	1.000

[1] Source: DCNO, OP-01.

APPENDIX E

COMPUTER PROGRAMS

```
      ∇BDISPLAY[⎕]∇
    ∇ BDISPLAY;S
     'ENTER SYMBOL OF VARIABLE TO BE DISPLAYED'
     '      TO QUIT TYPE 0'
     '  '
     '      B (BILLET REQUIREMENTS)'
     '      F (JOB SHARING FRACTIONS)'
     '      T (TARGET ALLOCATIONS)'
     '      X (CALCULATED BILLETS FILLED)'
     '      A (CALCULATED ALLOCATIONS)'
     '      E (PER CENT ERROR IN MEETING BILLET REQUIREMENTS)'
     '⍞'
    L0:→((S←⍞)='BFTXAE')/L1,L2,L2,L1,L2,L3
     →0
    L1:⍎'LABIJ ',S
     →L0
    L2:⍎'LABKIJ ',S
     →L0
    L3: ⍎'LABIJ ROUND BILPRCNT
     →L0
    ∇

      ∇BILPRCNT[⎕]∇
    ∇ BILPRCNT
     LABIJ ROUND 100×(X-B)÷B+B=0
    ∇

      ∇BINQP[⎕]∇
    ∇ Z←BINQP RTOT
     H←((⍉H1)+.×UU+.×H1),-0.5×K⍴1
     H←H,[1](K⍴1),0
     YL←(((⍉H1)+.×UU+.×,B),[1] RTOT)⌹H
     Y←K↑YL
     INVENTORY
     Z←PF1
    ∇

      ∇BINSEARCH[⎕]∇
    ∇ X←BINSEARCH P
    LOOP:→(EPS>|P[1]-P[2])/0
     X←(+/P)÷2
     →(0<BINQP X)/L1
     P[2]←X
     →LOOP
    L1:P[1]←X
     →LOOP
    ∇
```

```
      ∇CHANGE[⎕]∇
     ∇ CHANGE
[1]    'WHAT IS THE NAME OF THE VARIABLE TO BE CHANGED?'
[2]    X←⍞
[3]    →(X='0')/END
[4]    →((ρ,X)= 1 3 5)/C1,C2,C3
[5]    →ERR2
[6]   C2:⍎IF '×/X=''PHI''' THEN '→L' ELSE '→ERR2'
[7]   C3:⍎IF '×/X=''THETA''' THEN '→L' ELSE '→ERR2'
[8]   C1:→(1≠+/X='BPFGW')/ERR2
[9]   L:'WHAT IS THE LOCATION OF THE ELEMENT YOU WISH TO CHANGE?'
[10]  L0:→(0=×/(⍎'ρ,',X)≥KIJ←⎕)/ERR1
[11]   'INPUT NEW ',X,'(',(⍕KIJ),')'
[12]   INDEX←⍕KIJ[I←1]
[13]  L1:→((ρKIJ)<I←I+1)/OUT
[14]   INDEX←INDEX,';',⍕KIJ[I]
[15]   →L1
[16]  OUT:⍎X,'[',INDEX,']←⎕'
[17]   →END
[18]  ERR1:'INPUT ERROR. TRY AGAIN.'
[19]   →L0
[20]  ERR2:'SYMBOL NOT UNDERSTOOD'
[21]   'ENTER AGAIN OR TYPE 0 TO QUIT'
[22]   →1
[23]  END:'END OF CHANGE PROGRAM'
     ∇

      ∇CDIAG[⎕]∇
     ∇ Z←CDIAG X;IJ;J
[1]    IJ←ρX
[2]    Z←DG X[;J←1]
[3]   L1:→(IJ[2]<J←J+1)/0
[4]    Z←Z,[1] DG X[;J]
[5]    →L1
     ∇

      ∇CQP1Y[⎕]∇
     ∇ CQP1Y
[1]    'INPUT TOTAL NUMBER OF RECRUITS DESIRED'
[2]    RTOT←⎕
[3]    H←((⍉H1)+.×UU+.×H1),-0.5×Kρ1
[4]    H←H,[1](Kρ1),0
[5]    YL←((((⍉H1)+.×UU+.×,B),[1] RTOT)⌹H
[6]    Y←K↑YL
     ∇
```

```
      ∇DG[⎕]∇
    ∇ RR←DG X;N
[1]   N←⍴X
[2]   RR←(N,N)⍴0
[3]   RR[1;]←X
[4]   RR←(0,-⍳N-1)⌽[1] RR
    ∇
```

```
      ∇DISPIN[⎕]∇
    ∇ DISPIN;S
[1]   'ENTER SYMBOL OF VARIABLE TO BE DISPLAYED'
[2]   '       TO QUIT TYPE 0'
[3]   '       TO SEE MENU TYPE M'
[4]  L4:'  '
[5]   '       B (BILLET REQUIREMENTS)'
[6]   '       F (JOB SHARING ARRAY)'
[7]   '       T (TARGET ALLOCATIONS)'
[8]   '       G (PEOPLE SHARING ARRAY)'
[9]   '       P (PEOPLE REQUIREMENTS)'
[10]  '       W (EXPECTED FURTHER DURATION)'
[11] L0:'  '
[12]  '  '
[13]  '⍞'
[14]  →(1≠⍴S←⍞)/ERR1
[15]  →(S='BFTGPWM')/L1,L2,L5,L2,L6,L3,L4
[16]  →END
[17] L1:⍎'LABIJ ROUND ',S
[18]  →L0
[19] L2:⍎'LABKIJ ',S
[20]  →L0
[21] L3:⍎'LABIK ',S
[22]  →L0
[23] L5:⍎'LABKIJ ROUND ',S
[24]  →L0
[25] L6:⍎'LABIK ROUND ',S
[26]  →L0
[27] ERR1:'INCORRECT ENTRY'
[28] END:'END OF DISPLAY PROGRAM'
    ∇
```

```
      ∇IF[⎕]∇
    ∇ Z←IF A
[1]   Z←A
    ∇
```

```
      ∇THEN[⎕]∇
    ∇ Z99←A99 THEN B99
[1]   Z99←,B99[1+⍎A99;]
    ∇
```

```
      ∇ELSE[⎕]∇
    ∇ Z←A ELSE B;F
[1]   F←(⍴,A)⌈⍴,B
[2]   Z←(F↑B),[0.5] F↑A
    ∇
```

```
      ∇GWAITS[⎕]∇
    ∇ GWAITS;A;SF;N;I;AA;BB;DD;RR
      A←S
      SF←ALFA
      N←1↑ρSF
      I←ρA
      AA←A∘.≥⍳N
      BB←A∘.≥¯1+⍳N
      BB←BB∧~AA
      DD←⍉(N,I)ρA-⌊A
      AA←AA+DD×BB
      DD←(I,-I)↑(I,I+1)ρ¯1,1,Iρ0
      AA←DD+.×AA
      RR←AA+.×SF
      W←RR
      ∇
```

```
      ∇INVENTORY[⎕]∇
    ∇ INVENTORY
      Z←W×(I,K)ρY
      X←+/[1] A←G×⍉(J,I,K)ρZ
      ∇
```

```
      ∇LABIJ[⎕]∇
    ∇ Z←LABIJ X;N;IJ;I;J;L;HL;M;PP;ND;FS;H;XN;XX;NS
     ⍝THIS PROGRAM LABELS AN (I×J) ARRAY USING JLAB AND ILAB AS
     ⍝THE LABELS FOR ROWS AND COLUMNS RESPECTIVELY.  DP IS THE
     ⍝NUMBER OF DECIMAL PLACES TO BE DISPLAYED AND CS IS THE
     ⍝MINIMUM SPACING BETWEEN COLUMNS.
      N←0
      I←(ρX)[1]
      J←(ρX)[2]
      →(I≠(ρILAB)[1])/ERR1
      →(J≠(ρJLAB)[1])/ERR2
      L←(ρILAB)[2]
      HL←(ρJLAB)[2]
      M←(((2,L)ρ' '),[1] ILAB),[2]((I+2),2)ρ' |'
     L1:→(J<N←N+1)/L2
      PP←DP×0≠+/1|XX←|X[;N]
      ND←ρ⍕⌊XX[(⍒XX)[1]]
      NS←0≠+/(0>X[;N])
      FS←CS+HL⌈ND+PP+NS+1
      XN←(FS,PP)⍕(I,1)ρX[;N]
      H←((FS-HL)ρ' '),JLAB[N;]
      H←(2,ρH)ρH,(ρH)ρ'-'
      M←M,[2] H,[1] XN
      →L1
     L2:Z←M
      →0
     ERR1:'ERROR: NUMBER OF ROWS ≠ NUMBER OF ROW LABELS'
      →0
     ERR2:'ERROR: NUMBER OF COLUMNS ≠ NUMBER OF COLUMN LABELS'
      →0
      ∇
```

```
      ∇SSAM[⎕]∇
     ∇ SSAM
[1]   LIST1 MENU DC1
[2]   'END OF SSAM'
     ∇

      ∇FSTAR[⎕]∇
     ∇ F←FSTAR;I
[1]   F←(I,K,I,J)⍴I←0
[2]  L0:→(I<I←I+1)/L1
[3]   →L0,⍴F[I;;I;]←F[;I;]
[4]  L1:F←((I×K),(I×J))⍴F
     ∇

      ∇PECPRCNT[⎕]∇
     ∇ PECPRCNT
[1]   LABIK ROUND 100×(Z-P)÷P+P=0
     ∇
     .

      ∇PF1[⎕]∇
     ∇ E←PF1
[1]   E←+/+/(U×X-B)*2
     ∇
     .

      ∇PF2[⎕]∇
     ∇ D←PF2
[1]   D←+/+/(V×Z-P)*2
     ∇
     .

      ∇PF3[⎕]∇
     ∇ ED←PF3
[1]   ED←PF1+PF2
     ∇

      ∇QUAD[⎕]∇
     ∇ QUAD
[1]   ⍎DCQUAD[TYPE;]
[2]   INVENTORY
     ∇
     .

      ∇ROUND[⎕]∇
     ∇ RR←ROUND AA
[1]   RR←(⌊AA)+(AA-⌊AA)≥0.5
     ∇
```

```
      ∇INPUT[⎕]∇
     ∇ INPUT;K;KIJ;N;S;Z
[1]    'ENTER THE NUMBER OF MANPOWER TYPES, NUMBER OF RANKS, AND'
[2]    'NUMBER OF BILLET TYPES AS A VECTOR.'
[3]    KIJ←⎕
[4]    ' '
[5]    'TYPE THE SYMBOL OF THE VARIABLE YOU WISH TO INPUT'
[6]    '      TYPE 0 TO QUIT'
[7]    '      TYPE M TO SEE MENU AGAIN'
[8]   L:'  '
[9]    'B  BILLET REQUIREMENTS (I×J)'
[10]   'P  PEOPLE REQUIREMENTS  (I×K)'
[11]   'F  BILLET SHARING ARRAY  (K×I×J)'
[12]   'G  PEOPLE SHARING ARRAY  (K×I×J)'
[13]   'S  SURVIVOR FRACTIONS  (N×K)'
[14]   'R  RETENTION RATES  (N×K)'
[15]   '⊖  UNIT ERRORS IN TERMS OF BILLETS  (I×J)'
[16]   'ϕ  UNIT ERRORS IN TERMS OF PEOPLE  (I×K)'
[17]   ' '
[18]  L0:'  '
[19]   ' '
[20]   '⍞'
[21]   →(1<ρ,S←⍞)/ERR1
[22]   →(S≠'⊖')/L01
[23]   S←'THETA'
[24]   →L1
[25]  L01:→(S≠'ϕ')/L02
[26]   S←'PHI'
[27]   →L3
[28]  L02:→(S='MBPFGSR')/L,L1,L3,L2,L2,L4,L4
[29]   →END
[30]  L1:'INPUT EACH ROW OF ',S
[31]   ⍎S,'←MATIN KIJ[2 3]'
[32]   →L0
[33]  L2:K←0
[34]   Z←KIJρ0
[35]  L21:→(KIJ[1]<K←K+1)/L22
[36]   'FOR RANK ',⍕K
[37]   'INPUT EACH ROW OF ',S
[38]   Z[K;;]←MATIN KIJ[2 3]
[39]   →L21
[40]  L22:⍎S,'←Z'
[41]   →L0
[42]  L3:'INPUT EACH ROW OF ',S
[43]   ⍎S,'←MATIN ⊖KIJ[1 2]'
[44]   →L0
[45]  L4:'INPUT TOTAL NUMBER OF YEARS N'
[46]   N←⎕
[47]   'FOR EACH YEAR, INPUT EACH ROW OF ',S
[48]   ⍎S,'←MATIN N,KIJ[1]'
[49]   →L0
[50]  ERR1:'ERROR IN ENTRY OF VARIABLE'
[51]  END:'END OF INPUT PROGRAM'
     ∇
```

```
    ∇QP1Y[⎕]∇
   ∇ QP1Y
[1]  Y←((⍉H1)+.×UU+.×,B)⌹(⍉H1)+.×UU+.×H1
   ∇
   .

    ∇QP3Y[⎕]∇
   ∇ QP3Y
[1]  Y←(R+(⍉H1)+.×UU+.×,B)⌹Q+(⍉H1)+.×UU+.×H1
   ∇
   .

    ∇QP3XY[⎕]∇
   ∇ QP3XY;M;N;H;XYL
[1]  H←UU,(((M←I×J),K)⍴0),-0.5×⍉FSTAR
[2]  H←H,[1]((K,M)⍴0),Q,0.5×⍉WSTAR
[3]  H←H,[1](-FSTAR),WSTAR,(N,N←I×K)⍴0
[4]  XYL←((UU+.×,B),R,N⍴0)⌹H
[5]  Y←M↑(-N)↓XYL
   ∇
   .

    ∇QP2W[⎕]∇
   ∇ QP2W
[1]  Q←DG,(V*2)×(I,K)⍴Y
[2]  R←(V*2)×P
[3]  W←(I,K)⍴((⍉Q)+.×,R)⌹(⍉Q)+.×Q
   ∇

    ∇MENU[⎕]∇
   ∇ MENU MENU GOTO;XNUM;NUM;II
[1]   ' '
[2] L:'TYPE THE NUMBER OF THE PROGRAM(S) YOU WANT.'
[3]   ' '
[4]   MENU
[5]   XNUM←⍴NUM←,⍞
[6]   II←0
[7] L0:→(XNUM<II←II+1)/L
[8] L1:⍎IF 'NUM[II]∊(⍴MENU)[1]↑''0123456789''' THEN '→L2' ELSE '→L0'
[9] L2:⍎IF NUM[II] THEN GOTO
[10]  →L0
   ∇
```

```
      ∇MAKEBF[⎕]∇
    ∇ MAKEBF
[1]   B←+/[1] T←G×⍉(J,I,K)⍴P
[2]   F←T÷(K,I,J)⍴B+B=0
    ∇
    .

      ∇MAKEFP[⎕]∇
    ∇ MAKEFP
[1]   P←(I,K)⍴F←(K,I,J)⍴I←0
[2]  L1:→(I<I←I+1)/0
[3]   P[I;]←(G+.×B←B[I;])⌹(G+.×⍉G←G[;I;])
[4]   →L1,⍴F[;I;]←G×(⍉(J,K)⍴P[I;])÷(K,J)⍴B+B=0
[5]   T←F×(K,I,J)⍴B
    ∇
    .

      ∇MAKEPG[⎕]∇
    ∇ MAKEPG
[1]   P←⍉+/[3] T←F×(K,I,J)⍴B
[2]   G←T÷⍉(J,I,K)⍴P+P=0
    ∇
    .

      ∇MATIN[⎕]∇
    ∇ Z←MATIN IJ;I;T
[1]  ⍝ THIS PROGRAM IS USEFUL FOR MATRIX INPUT. THE SYNTAX IS :
[2]  ⍝        X← MATIN 5 8
[3]  ⍝ X IS THE MATRIX TO BE FORMED AND HAS DIMENSION (5×8).
[4]  ⍝ THE PROGRAM PROMPTS FOR THE REST.
[5]   Z←⍳I←0
[6]  L1:→(IJ[1]<I←I+1)/L4
[7]  L2:'INPUT ROW ',⍕I
[8]   →(IJ[2]=⍴T←⎕)/L3
[9]   'INPUT ERROR: EACH ROW HAS ',(⍕IJ[2]),' ELEMENTS'
[10]  →L2
[11] L3:Z←Z,T
[12]  →L1
[13] L4:Z←IJ⍴Z
    ∇
    .

      ∇MENU[⎕]∇
    ∇ MENU MENU GOTO;XNUM;NUM;II
[1]   ' '
[2]  L:'TYPE THE NUMBER OF THE PROGRAM(S) YOU WANT.'
[3]   ' '
[4]   MENU
[5]   XNUM←⍴NUM←,⎕
[6]   II←0
[7]  L0:→(XNUM<II←II+1)/L
[8]  L1:⍎IF 'NUM[II]∊(⍴MENU)[1]↑''0123456789''' THEN '→L2' ELSE '→L0'
[9]  L2:⍎IF NUM[II] THEN GOTO
[10]  →L0
```

```
      ∇WAITS[⎕]∇
     ∇ STAGE WAITS CON;A1;MJ;M;J;M1;M2;W1;I
[1]   J←(ρA1←×\CON)[2]
[2]   M1←(Jρ1),[I←1] A1
[3]   M2←A1,[1] Jρ0
[4]   W1←(¯1,0)↓M2+(M1-M2)÷2
[5]   STAGE←0,STAGE
[6]   W←(0,J)ρι0
[7]  L1:→((ρSTAGE)<I←I+1)/0
[8]   →L1,ρW←W,[1]+⌿(STAGE[I-1],0)↓((STAGE[I],J)↑W1)
     ∇

      ∇UU[⎕]∇
     ∇ Z←UU
[1]   Z←DG,U*2
     ∇
     .

      ∇V[⎕]∇
     ∇ V←V
[1]   V←(P≠0)÷(PHI÷100)×P+P=0
     ∇
     .

      ∇WSTAR[⎕]∇
     ∇ W←WSTAR
[1]   W←CDIAG⍉W
     ∇

      ∇H1[⎕]∇
     ∇ Z←H1
[1]   Z←⍉(K,I×J)ρG×⍉(J,I,K)ρW
     ∇
     .

      ∇Q[⎕]∇
     ∇ Z←Q
[1]   Z←DG+/[1](V×W)*2
     ∇
     .

      ∇R[⎕]∇
     ∇ Z←R
[1]   Z←+/[1] P×W×V*2
     ∇
     .

      ∇U[⎕]∇
     ∇ U←U
[1]   U←(B≠0)÷(THETA÷100)×B+B=0
     ∇
```

LIST OF REFERENCES

1. Grinold, R. C., Naval Postgraduate School Report NPS55-79-025, A Steady State Longitudinal Manpower Planning Model With Several Classes by Manpower, August 1979.

2. Grinold, R. C., and K. T. Marshall, Manpower Planning Models, North-Holland, 1977.

3. Bartholomew, D. J. and A. Forbes, Techniques of Manpower Modelling, Wiley, 1979.

4. Department of the Navy, NAVPERS 15197A, Unrestricted Line Officer Career Guidebook, 1979.

5. Lee, S. M., Goal Programming for Decision Analysis, Auerbach, 1972.

6. Hillier, F. S. and G. J. Lieberman, Introduction to Operations Research, 3rd ed., Holden-Day, 1980.

7. Luenberger, David G., Introduction to Linear and Nonlinear Programming, Addison-Wesley, 1973.

8. Duff, Richard H., A Microcomputer-Based Network Optimization Package, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1981.

9. Simmons, Donald M., Nonlinear Programming for Operations Research, Prentice-Hall, 1975.

BIBLIOGRAPHY

Bartholomew, D. J. and Forbes, A., Techniques of Manpower Modelling, Wiley, 1979.

Bres, E. S., Burns, D., Charnes, A., and Cooper, W. W., "A Goal Programming Model for Planning Officer Accessions," Management Science, v. 26, no. 8, August 1980.

Department of the Navy, NAVPERS 15197A, Unrestricted Line Officer Career Guidebook, 1979.

Duff, Richard H., A Microcomputer-Based Network Optimization Package, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1981.

Gilman, L., and Rose, A. J., APL: An Interactive Aapproach, Wiley, 1976.

Grinold, R. C., A Steady State Longitudinal Manpower Planning Model With Several Classes by Manpower, Naval Postgraduate School Report NPS55-79-025, August, 1979.

Grinold, R. C., and Marshall, K. T., Manpower Planning Models, North-Holland, 1977.

Hillier, F. S. and Lieberman, G. J., Introduction to Operations Research, 3rd ed., Holden-Day, 1980.

Holt, C. H. and others, Planning Production, Inventories, and Work Force, Prentice-Hall, 1960.

Lee, S. M., Goal Programming for Decision Analysis, Auerbach, 1972.

Lee, S. M., Franz, L. S., and Wynne, A. J. "Optimizing State Patrol Manpower Allocation," Journal of the Operational Research Society, v. 30, no. 10, 1979.

LePage, W. R., Applied APL Programming, Prentice-Hall, 1978.

Luenberger, David G., Introduction to Linear and Nonlinear Programming, Addison-Wesley, 1973.

Martel, A. and Price, W., "Stochastic Programming Applied to Human Resource Planning," Journal of the Operational Research Society, v. 32, no. 3, 1981.

Polivka, R. P. and Pakin, S., APL: The Language and Its Usage, Prentice-Hall, 1975.

Ramsay, J. B. and Musgrave, G. I., APL-STAT, Lifetime Learning, 1981.

Simmons, Donald M., Nonlinear Programming for Operations Research, Prentice-Hall, 1975.

Zanakis, S. H. and Maret, M. W., "A Markovian Goal Programming Approach to Aggregate Manpower Planning," Journal of the Operational Research Society, v. 32, no. 1, 1981.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93940	2
3	Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, CA 93940	3
4.	Defense Logistics Studies Information Exchange U.S. Army Logistics Management Center Fort Lee, VA 23801	1
5.	Dean of Research Code 012 Naval Postgraduate School Monterey, CA 93940	1
6.	Director, Total Force Planning Division (OP-11) Office of DCNO (MPT) Department of the Navy Washington, D.C. 20370	1
7.	Director Military Personnel/Training Division (OP-13) Office of DCNO (MPT) Department of the Navy Washington, D.C. 20370	1
8.	Commandant, U.S. Marine Corps (Code RD & S-40) HQ, USMC Washington, D.C. 20380	1
9.	Mr. E. C. Grayson Office of DASN (Manpower) Department of the Navy Washington, D.C. 20350	1
10.	Mr. Joe Silverman (Code 303) Naval Personnel R&D Center San Diego, CA 92152	1

11.	LCDR William D. Ferree OP-132C Office of DCNO (MPT) Arlington Annex Washington, D.C. 20370	3
12.	Professor Richard C. Grinold Department of Business Administration University of California Berkeley, CA 94720	1
13	Mr. Greg Jenkins AD/XRSP Eglin AFB, FL 32542	1
14.	MAJ Richard H. Duff, USMC 9 Revere Road Monterey, CA 93940	1
15.	Dr. Lawrence Low Ketron, Inc. 900 Cherry Avenue, Suite 230 San Bruno, CA 94066	1
16.	Professor Paul R. Milch, Code 55Mh Naval Postgraduate School Monterey, CA 93940	10
17.	Dr. T. C. Varley (Code 434) Office of Naval Research 800 N. Quincy Street Arlington, VA 22217	1
18.	Mr. Robert K. Lehto (OP-16) Director, Total Information Systems Management Office of DCNO (MPT) Arlington Annex Washington, D.C. 20370	1
19.	Professor Richard Elster, Code 54Ea Naval Postgraduate School Monterey, CA 93940	1
20.	Professor Michael Sovereign, Code 55Zo Naval Postgraduate School Monterey, CA 93940	1
21.	Professor George Thomas, Code 54Te Naval Postgraduate School Monterey, CA 93940	1

22.	Dean of Information and Policy Sciences, Code 01 Naval Postgraduate School Monterey, CA 93940	1
23.	Department Chairman, Code 54 Department of Administrative Science Naval Postgraduate School Monterey, CA 93940	1
24.	Captain Joseph N. Lott, USMC 102 Malloway Lane Monterey, CA 93940	2

An investigation of a steady state, allocation model for military manpower planning.